Exchange 2000 QuickDoc
Quick Tuning Guide

QD205 **Version 1.2**

Compiled by:
Paul Bowden, Program Manager
Microsoft Exchange Server Product Unit

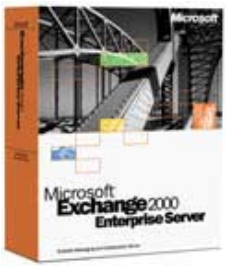
© 2000 Microsoft Corporation. All rights reserved.

The information contained in this White Paper represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

BackOffice logo, Microsoft, Win32, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other product or company names mentioned herein may be the trademarks of their respective owners.

Microsoft Corporation • One Microsoft Way • Redmond, WA 98052-6399 • USA



Tuning Overview

The purpose of this document is to provide advanced Exchange administrators with information on how to tune an Exchange 2000 server based on its local environment. There are many variables involved with performance tuning. What may work well on some servers, may not effect other servers.

The aim of this document is to outline the major areas that may require tuning. This document does not cover extensive performance analysis, instead it concentrates on the 'quick wins'. That is, performance tuning changes that are easy to identify. This usually involves changing settings in the registry. If you are unfamiliar with the registry and how to change it, consult an expert. Even if you are very familiar with the registry, you should always carefully document the changes that you make, and monitor your system after each change. You should also tune your system in stages. If you make many changes all at once, and the performance of your server degrades, you won't know which change had the negative effect and you will need to back out all changes.

Keep all tuning alterations in a master document and make your team aware of them. Additionally, supply your tuning document to product support services if you have a need to call them regarding your servers.

Finally, if your server is not exhibiting performance problems, then you should be cautious in making performance tuning adjustments.

IMPORTANT: This article contains information about editing the registry. Before you edit the registry, make sure you understand how to restore it if a problem occurs. For information about how to do this, view the "Restoring the Registry" Help topic in Regedit.exe or the "Restoring a Registry Key" Help topic in Regedt32.exe.

WARNING: Using Registry Editor incorrectly can cause serious problems that may require you to reinstall your operating system. Microsoft cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use Registry Editor at your own risk.

For information about how to edit the registry, view the "Changing Keys and Values" Help topic in Registry Editor (Regedit.exe) or the "Add and Delete Information in the Registry" and "Edit Registry Data" Help topics in Regedt32.exe. Note that you should back up the registry before you edit it. If you are running Windows NT or Windows 2000, you should also update your Emergency Repair Disk (ERD).

Changes in the tuning philosophies from Exchange 5.5 to Exchange 2000 Server

Performance in Exchange 5.5 could be tuned by the administrator using the Microsoft Exchange Server Performance Optimizer. The Optimizer (a.k.a. "PerfWiz") allowed the administrator to specify how the server would be configured (Private Store, Public Store, Connectors, POP/IMAP, etc.), if its memory usage should be limited, and how many users it would be expected to handle. Based on the administrator's choices, files written by the various Exchange components (Information Store, MTA, etc.) could be assigned to specific fixed disks depending upon available storage.

Exchange Server 2000 does not use a Performance Optimizer-like tool. One reason for this is that the new release of Exchange is better capable of dynamically changing certain parameters, spinning-up more threads, etc., as needed. To go beyond these dynamic changes, the administrator is responsible for manually optimizing disk utilization and for manually modifying registry keys, but fewer such registry changes are necessary.

Customer variation is probably the greatest variable in tuning Exchange for optimum performance. Even customers with similar needs will choose solutions that differ slightly or significantly. Hardware

varies in the number and speed of the processors, the available RAM, and the number of disks and the disk configuration chosen (RAID level). Exchange can be configured with different numbers of Storage Groups and databases, and may or may not be clustered with other servers accessing a central storage subsystem. The server load will vary due to the total number of users with mailboxes on the server, the number of users logged-on at a given time and the actions they are performing, and any additional load imposed by routing of outside messages through the server.

In Exchange 2000, features that can be optimized include: Disks, Message Transfer Agent, SMTP, Information Store Database and Extensible Storage Engine Cache and Log Buffers, Active Directory Connector, Active Directory Integration, IFS Handle Cache, Credentials Cache & Mailbox Cache, DSAccess Cache, DSProxy, and POP and IMAP settings.

Tuning Considerations

The efficiency and capacity of a Microsoft Exchange 2000 Server installation depends on the administrator's choices of server and storage hardware, and of the installation's topology, and these factors in turn should be chosen based on expected types and levels of usage. Further efficiency can be achieved by changing various registry settings on the Exchange server machine.

There are three main types of tuning parameters: those with fixed optimal values (or values that can be treated as such), those that can be dynamically tuned by the software, and those that must be manually tuned (via Setup, the Exchange System Manager, the registry, and ADSI Edit).

Parameters may need to be manually tuned for several reasons. Hardware or Exchange configuration information may be needed and this information can't (or won't) be obtained dynamically. Server load information may also be required; this can't be obtained dynamically either.

Upgrading from Exchange 5.5 to Exchange 2000

When Exchange 5.5 is upgraded to Exchange 2000, some registry keys altered by PerfWiz retain their PerfWiz values, some do not, and some keys no longer appear in the registry or appear in a different location. Both because of the significant improvements made to Exchange 2000, and because of optimizations that do not necessarily carry over from Exchange 5.5, it is best to assume that one should start over in evaluating optimization of Exchange 2000. Knowing your Exchange 5.5 settings before upgrading to Exchange 2000 will be useful in this respect. The text file *WINNT\System32\perfopt.log* provides a record of those registry keys and disk assignments changed by PerfWiz.

Hardware Configuration

The hardware that you choose for your Exchange deployment will have the greatest impact on the performance that you receive. It's difficult to quote machine specifications and the resulting performance, as there are too many variables. However, it is possible to match hardware components against the Exchange 2000 services, and in particular, call out the resource intensive processes.

Processor

It is true to say that Exchange 5.5 is not processor intensive. The base services do not consume masses of processor time. With Exchange 2000, there are three services that can place a significant load on the processor – Search, Conferencing and Front-end configurations.

Search – Full text indexing is a great new facility in Exchange 2000. It allows clients and applications to search through gigabytes of Exchange data and receive lightning fast results. However, the result list returned to the client is accurate up to the time when the last crawl took place. Ideally, it is good to perform a crawl every hour and rebuild the catalog every night. When the MSSearch service is crawling and rebuilding it will place a constant load on the processor, usually you will see the processor at a rate of 60% or higher. Because of this, consider fast or multiple processors when turning on full text indexing. Ultimately, if the processor in the Exchange server is too slow, catalog updates will be slow and users may complain that they cannot search on recently added items.

If you are only interested in indexing Public Folder data, you may wish to home those folders to a fast, dedicated server within the Admin Group.

Conferencing – There are three services that make up the Exchange Conferencing Server product. The Conference Management Service (CMS) is responsible for the management of virtual rooms and client connection point. The T.120 Multi-point Control Unit (MCU) provides the data conferencing facility; it is responsible for taking unicast data, such as shared whiteboard and chat from client computers and re-distributing it to other clients and servers participating in the conference. The H.323 bridge service provides a mechanism where unicast clients can stream audio and video to the server and get this bridged into the conference. You would usually enable H.323 support if your clients and/or network does not support the IP multicast protocol.

The CMS places an insignificant amount of load on the processor. Depending on how many data conferences are running and how many clients are connected, the T.120 service can be processor intensive. However, by far and away the most intensive process in the conferencing suite is the H.323 bridge. Without H.323, clients are responsible for streaming audio and video to one another – directly. With H.323, all this data has to be funneled through the server and passed back out to other clients and servers. Although the conferencing traffic will be 'bursty', if the processor installed within the server is too slow, users will notice an inferior quality of the audio and video.

If you are using H.323 services, you may decide to deploy this component onto a separate high-speed server.

Front-end server – The role of an Exchange front-end server is to receive incoming HTTP, POP and IMAP requests, interpret the protocol verbs, perform lookups in the Active Directory, and then proxy the request to a back-end. Front-end servers are expected to handle thousands of client connections. Due to the role of the front-end server, it will be more processor rather than memory or disk intensive.

Memory

By default, Exchange 2000 uses all of the physical memory available (if you wish to also run other software on your servers, you can restrict the amount of memory used by Exchange). The biggest individual consumer of memory in Exchange 2000 is the store process. On an active, production Exchange 2000 server it is not uncommon to notice that the store is consuming nearly all of the server memory. Like Exchange 5.5, the store uses a unique cache mechanism called Dynamic Buffer Allocation (DBA). This process self-governs how much memory it uses, and balances that with other applications running on the server. If Exchange 2000 is the only application running, DBA will decide to allocate more memory to itself.

The amount of memory that you need in your server depends on the number of databases, size of databases and number of transactions. It's true to say that as you create more Exchange databases on the server, your memory requirement increases sharply. You can do a certain amount of memory preservation by being clever with your database configuration. The first database in a storage group consumes the greatest amount of virtual memory, whereas if you add a new database to an existing storage group, the memory consumption of this database will be less.

Exchange 2000 can handle up to 20 databases per server, this is made up from a maximum of 4 storage groups and 5 databases per storage group. Wherever possible, you should fill out your storage groups to the maximum number of databases before creating a new group. The advantages to doing this are reduced memory consumption and disk overheads, however, there are a few trade-offs to make:

- Circular logging can only be controlled at the storage group level
- If one database in the group goes offline because of an error, the other databases in that group temporarily go offline and remount (note that this scenario does not apply to out of disk space errors)
- Only one backup process can take place in a single Storage Group at any one-time
- Backing up one database in a storage group will halt the online maintenance of all other databases in the storage group

If your Exchange server has 1Gb of memory or greater, it's very important to add the /3GB switch to the boot.ini file on the server. By default, Windows 2000 Advanced Server reserves 2Gb of virtual address space for the kernel, and allows user mode processes (such as the Exchange 2000 store) to use 2Gb of virtual address space. Virtual address space for a specific process is allocated at startup and increases as more memory is used during run-time. It is normal for the actual memory usage (working set) of a process to be much less than the address space the process was allocated. On an Exchange 2000 server with 1Gb or more of memory, it is necessary to modify Windows 2000 Advanced Server so that 3Gb are available for user mode applications.

It is very important that the Store.exe process does not run out of virtual address space. When this happens, memory allocations will fail (even if there is plenty of physical RAM left) and you must restart the Information Store service. For example, a server with 2Gb of physical RAM without the /3GB switch in the boot.ini file will run out of memory when the Store.exe's virtual address space reaches 2Gb. Windows Task Manager will show that only about 1.5Gb is actually being used but the server will be out of memory nonetheless. For more information on how to set the /3GB switch, refer to knowledge base article Q266096.

Network Interface

Much of the network interface subsystem will be auto-tuned. Server-based network cards and their drivers are very smart at intelligently detecting the type and level of traffic passing through the network interface, and will self-tune to reflect this information. Beyond ensuring that you have the latest device driver on the server, there is not much to do here.

If you intend on configuring IPSec on your Exchange 2000 server, perhaps because of a front-end / back-end configuration, you may want to invest in a specialized network card that can offload IPSec processing from the main CPU. For example, both Intel and 3Com produce cards that offer this feature. For more information, see the vendors web-site:

- http://www.intel.com/network/products/pro100ssrvr.htm
- http://www.3com.com/products/nics/3cr990.html

Disk Subsystem

Ultimately, your Exchange data has to make its way down to the disk, and this is one of the most important areas of configuration. Probably the most asked question is "I have xx disks in my server, what is the best way to partition them?". Before we can answer this question, you have to consider the role of the server – for example, tuning a bridgehead will be different to tuning a mailbox server.

SMTP Bridgehead – If the server is an SMTP bridgehead, then generally the best disk layout is to create one partition. Messages will arrive on the SMTP interface, written into the mailroot folder (which is in NTFS) and then passed to the next computer. The best performance will be gained by spanning the mailroot folder over as many disks as possible. Your hardware RAID configuration will be largely dictated by the number of disk spindles available:

Disks	***Recommended Hardware RAID level and partitioning***
2	RAID1 – Mirroring – One partition
3 – 5	RAID5 – Distributed Parity – One partition
6 (or more)	RAID0+1 – Data striping and Mirroring – One partition

If your hardware RAID controller has a mirrored, battery-backed cache, and allows you to tune the read/write cache ratio, you should set it to 100% write. The reasoning behind this is that the server can only acknowledge receipt of a message once it has been locked down to disk, therefore the faster the server can write to disk, the more responsive the server will be to other computers. Setting the cache to 100% write is especially important for RAID5 configurations where one application I/O means four disk I/Os. Although it is true that the message contents need to be re-read before being relayed to the next computer, the SMTP service will have an open file handle to the data and will retrieve the contents from the NT file system cache. Setting the controller for 100% write will also speed up SMTP service shut-down, this is useful if you increase the SMTP file handles (discussed later).

X.400 Bridgehead or legacy Connector server – If your Exchange 2000 server houses many X.400 Connectors, or connects to legacy messaging systems such as Lotus cc:Mail, Notes, Novell GroupWise or Microsoft Mail, then you should consider creating a separate disk partition for transaction logs if you have enough spindles. Messages that use these Exchange components will move data through the store process, and if the server is heavily loaded, additional performance will be gained by splitting the transaction logs from the database.

When a message is received by the MTA, the data is written into the mtadata folder on an NTFS partition and passed into the store via a MTS-IN “virtual” queue. Because the message hits the store, data will be written to the transaction log files. The message then goes through the categorization and routing process. If routing has determined that the message should be sent out through an X.400 Connector, the data will be moved into the MTS-OUT “virtual” queue. Legacy messaging connectors use a similar process.

Your hardware RAID configuration will be largely dictated by the number of disk spindles available:

Disks	***Recommended Hardware RAID level and partitioning***
2	RAID1 – Mirroring – One partition
3 – 4	RAID5 – Distributed Parity – One partition
5	RAID5 (C:) for three disks - binaries, RAID1 (D:) for two disks – logs
6	RAID5 (C:) for three disks - binaries, RAID1 (D:) for two disks – logs, No RAID (E:) for sixth disk – Page file

If your hardware RAID controller has a mirrored, battery-backed cache, and allows you to tune the read/write cache ratio, you should set it to 100% write for configurations that use a single partition, and 100% write on the transaction log partition when five or more disks are used. You will see a better performance gain by using write cache on the transaction logs rather than read cache on the database drive.

Mailbox and Public Folder servers – If you are configuring a single database collaboration server your disk design will be similar to that of an Exchange 5.5 server. The number one priority is to split the transaction logs and database onto separate disk spindles. It must be remembered that the primary reason for the split is to gain performance on transaction logging, and take advantage of sequential writes. Not only should the logs be on a different spindle, but that spindle should be dedicated to logs

alone. For example, you will not receive the performance benefits of the split if you choose to place the page file or Windows binaries on the same spindle as the logs.

If you intend to create multiple storage groups on the server, then again your number one priority is to split the logs onto dedicated spindles. However, to gain any performance, you must have a dedicated spindle set for each storage group. Once you've achieved this state, it's time to work out how many spindles you have left. On the majority of PC servers, you won't have that many disk slots available, so your best option may be to place all databases onto the same partition. However, there are good reasons for placing database files from different storage groups onto separate partitions:

a. It makes management easier and more logical

b. If the drive array housing one set of databases goes off-line, then it won't effect databases in other storage groups.

To expand on the second point, if you spread databases from multiple storage groups over the same array and that array goes down, ALL databases from those storage groups (even those databases located on other arrays) will temporarily go offline. The bad database will be marked as down, and the good databases will remount. The outage should only last for a few minutes, and MAPI clients such as Outlook will recover their connections.

With enough disks, you could go to the extent of splitting EDB files from the STM files. The performance benefit of doing this will really be dictated by the types of client that your server accommodates.

MAPI and OWA clients will read and write data to and from the EDB file. Technically, clients communicate to the store, so they have no understanding of EDB vs. STM. All other clients, will use the STM file. The table below indicates the *preferred* location used by different clients.

Client Type	***Submitting data***	***Retrieving data***
Outlook in MAPI mode	EDB	EDB
POP	STM (via SMTP)	STM
IMAP	STM (via SMTP)	STM
OWA	EDB	EDB
IFS	STM	STM
SMTP	STM	Not applicable
SMTP (with MAPI data)	STM (but full promotion to EDB)	Not applicable
ADO / OLEDB	STM	STM
CDO for Exchange 2000	STM	STM
CDO 1.21	EDB	EDB

We say *preferred* file type as cross conversions can and do occur. For example, if a POP3 client submits a message, the message data is physically stored in the STM. There is some property promotion to the EDB of the message header and other data. This takes place automatically. However, now assume that a MAPI client attempts to read the message. In this scenario, the message data in the STM is fully promoted to the EDB file (attachments are left in the STM and "converted" in-memory). Promotion only happens from the STM to the EDB. There are no circumstances where data is promoted to the STM, all client conversions are performed in-memory in these scenarios.

In the majority of scenarios, there is no real benefit to splitting the EDB and STM files onto separate spindles. For example, if you have six disks, you have to ask yourself whether creating two RAID5 partitions over three disks each would give more performance than a single RAID0+1 (stripe and mirror) partition. The answer is that RAID0+1 will always give better performance.

It should be noted that specialized storage subsystems (such as EMC) that employ gigabytes of cache and tens of spindles usually have their own partition management and fault-tolerant operating system. When working with these devices, it is normal to place all data onto the same disk partition and let the hardware perform the optimizations.

In summary, your golden rules for configuring collaboration servers are as follows:

1. Create a RAID1 partition for Windows and Exchange binaries
2. Place the pagefile on a separate spindle (this can be coupled with the binary drive on smaller servers)
3. Create one dedicated fault-tolerant partition per storage group for the transaction logs (RAID1 or RAID0+1)
4. Create at least one fault-tolerant partition for your databases. If you only have one array, place all databases on this array. If you have multiple arrays, use one array per storage group (databases only)

Databases and transaction log files can be moved to different partitions by using the Exchange System Manager. More information on this subject can be found in knowledge base article Q257184.

Front-end servers – Front-end servers are not disk intensive. Indeed, on a 'true' front-end server, you should delete the databases through the Exchange System Manager and stop the Store process. This will free up valuable processing time for the INETINFO process. There are exceptions to the rule of deleting the stores. For example, some companies will want a front-end to accept SMTP messages from clients, and in this scenario the Store is used for tasks such as NDR generation. For more information on configuring and optimizing front-end servers, refer to the whitepaper at http://www.microsoft.com/exchange/techinfo/E2KFrontBack.htm

For more information, you may wish to read the excellent Exchange 2000 Performance Planning article at:
http://www.win2000mag.com/Articles/Index.cfm?ArticleID=8287&SearchText=performance%20and%20exchange%202000

Message Transfer Agent Tuning

Exchange 2000 does not include a Performance Optimization wizard mainly because the majority of Exchange 2000 components are self-tuning. However, when the message transfer agent (MTA) is installed, its tuning state reflects that of an Exchange 5.5 server that has never been performance optimized.

In scenarios where only Exchange 2000 servers exist in an organization, the MTA does not perform any processing, and thus does not need to be performance tuned. However, when you are co-existing between Exchange 5.5 and Exchange 2000, the MTA is used for all intra-site communications, and possibly between sites if RPC Site Connectors and/or X.400 Connectors are deployed. The MTA also processes messages coming from or going to legacy messaging systems such as Lotus cc:Mail, Notes, Novell GroupWise and Microsoft Mail.

In smaller organizations, the MTA process will not need to be manually tuned. For larger companies that encompass many servers in the same Site, many Sites within the Organization, or legacy messaging connectors, you should consider tuning the MTA registry parameters. You should also remember that the Exchange 2000 MTA places a much greater load on system resources (CPU, memory and disk) than the Exchange 5.5 MTA. You should take this into consideration when sizing your hardware. In a large mixed Exchange 5.5 and 2000 site, you may wish to divide your servers into different Routing Groups, and dedicate one of the Exchange 2000 servers to handling the 2000 to 5.5 MTA communication. The other Exchange 2000 servers in the site will use this dedicated bridgehead to send messages to the Exchange 5.5 servers, which will reduce MTA processing overhead.

The rest of this section details the registry parameters that might require adjustment and gives examples as to where tuning may be appropriate. It must be remembered that increasing the number of threads for a process does not always increase performance (it may actually decrease performance). A good example of this is the relationship between *RTS threads* and *Kernel threads* in the MTA. RTS threads are responsible for putting data in the queues, Kernel threads take the data out maneuvers it off-server. If the RTS threads setting is too high, local queues will fill up quickly and may starve the Kernel threads from reading the queue data, thus causing a message transfer slowdown.

MSExchangeMTA \ Parameters

The registry parameters in this section can be found under HKEY_LOCAL_MACHINE \ SYSTEM \ CurrentControlSet \ Services \ MSExchangeMTA \ Parameters

Parameter:	DB data buffers per object
Explanation:	This value is the number of database server buffers that are configured for each database object. More buffers require more memory, but make it less likely for a database object to be rolled out to disk because of a lack of buffer space
Default setting:	0x3
When to change:	If this MTA communicates with multiple Exchange 5.5 servers either within a site, or between sites. Additionally, you should tune if a legacy messaging connector is homed on this server
Recommend setting:	0x6

Parameter:	Dispatcher threads
Explanation:	This value is the number of MTA Dispatcher threads, which are responsible for the processing of messages. This is multiplied by three for the three subtypes (Router, Fanout, Result) of Dispatcher thread
Default setting:	0x1
When to change:	If this MTA communicates with other Exchange 5.5 servers. Additionally, consider tuning if more than five Exchange 5.5 servers exist in the site, or if Groups (i.e. Distribution Lists) are heavily used
Recommend setting:	0x3

Parameter:	Kernel threads
Explanation:	This value is the number of platform threads that handle the Presentation and Session level of the Open Systems Interconnection (OSI) stack. These threads are at the heart of MTA message processing
Default setting:	0x1
When to change:	If this MTA communicates with other Exchange 5.5 servers using RPC over slow or highly latent network connections
Recommend setting:	0x3 - Standard recommendation 0x8 - If the Exchange 2000 MTA belongs to a site containing more than 15 Exchange 5.5 servers 0xC (12) - If the Exchange 2000 MTA belongs to a site containing more than 30 Exchange 5.5 servers

Parameter:	Max RPC Calls Outstanding
Explanation:	This value is the maximum number of remote procedure call (RPC) threads. This limits the maximum number of RPC procedure calls that are guaranteed to be processed at one time
Default setting:	0x32 (50)
When to change:	Adjust if you increase the number of Gateway In/Out threads in the store (which is recommended in Exchange 5.5 and 2000 co-existence scenarios)
Recommended setting:	0x80 (128)

Parameter:	MDB users
Explanation:	Defines the number of DNs to cache from the directory
Default setting:	0x1F4 (500)
When to change:	Adjust when your organization contains more than 1500 users. Change the value to one third the size of the Global Address List, to a maximum of 5000
Recommended setting:	0x1388 (5000)

Parameter:	RTS threads
Explanation:	Reliable Transfer Service threads. This value is the number of platform threads that handle the RTSE level of the Open Systems Interconnection (OSI) stack
Default setting:	0x1
When to change:	If this MTA communicates with multiple Exchange 5.5 servers either within a site, or between sites
Recommended setting:	0x3

Parameter:	TCP/IP control blocks
Explanation:	This value is the maximum number of concurrent RFC1006 (TCP/IP) connections that are supported. This setting controls the number of buffers available for X.400 connections
Default setting:	0x14 (20)
When to change:	If hosting more than one X.400 Connector on the server
Recommended setting:	10 control blocks for each hosted X.400 Connector, plus 1 control block for incoming connections

Parameter:	Transfer threads
Explanation:	This value is the number of MTA Transfer threads. This is multiplied by two for the two subtypes (Transfer In, Transfer Out) of Transfer thread
Default setting:	0x1
When to change:	If this MTA communicates with multiple Exchange 5.5 servers either within a site, or between sites
Recommended setting:	0x3

You will need to restart the MTA before registry adjustments take effect.

MSExchangeIS

When messages are received into the MTA from an Exchange 5.5 server or legacy gateway, they are handed off to the Store process, and then up into the Advanced Queuing (A/Q) engine. In environments where many messages are arriving into the MTA simultaneously, it is advantageous to increase the number of processing threads between the Store and MTA. You can monitor MTA to Store queue build-ups using either Performance Monitor or the Exchange System Manager.

The following parameters need to be added under *HKEY_LOCAL_MACHINE \ SYSTEM \ CurrentControlSet \ Services \ MSExchangeIS \ <server-name> \ Private-<database-guid>*

Parameter:	Gateway In Threads (REG_DWORD)
Explanation:	Defines the number of threads available for retrieving messages from the MTA process into the Store
Default setting:	Not present, but defaults to 0x1
When to change:	If this MTA communicates with multiple Exchange 5.5 servers or legacy messaging connectors
Recommended setting:	0x3
Caveat::	Each of these threads will consume about 1Mb of virtual memory (VM). Further, the actual number of threads created will be this value multiplied by the number of databases. This could be an issue on servers with a large number of private databases. So if you have, say, 10 private databases and increase this parameter and the one below from 1 to 3 (a total increase of 4 threads), you will actually create 4 x 10 = 40 threads, which altogether will consume 40 Mb of VM

Parameter:	Gateway Out Threads (REG_DWORD)
Explanation:	Defines the number of threads available for sending messages from the Store to the MTA process
Default setting:	Not present, but defaults to 0x1
When to change:	If this MTA communicates with multiple Exchange 5.5 servers or legacy messaging connectors
Recommended setting:	0x3

You should add these parameters to all private databases configured on the server. After making this change, you must increase the “Max RPC Calls Outstanding” registry parameter for the MTA process (see previous section). This is covered by knowledge base article Q264075.

MTA File Directories

By default, the Exchange MTA database and run directories are located under the path where Exchange 2000 is installed (*<drive>*:\Program Files\Exchsrvr\mtadata). On some servers, especially where Exchange is acting as a bridgehead, you will get a positive performance impact by locating the MTA database onto a fast disk array, such as a RAID5 partition. You should not attempt to relocate the MTA run directory as this can cause undesirable side effects.

To re-locate the MTA database, use the following steps:

1. Stop the Microsoft Exchange MTA Stacks service. To do so, at the command-prompt type the following command: "net stop msexchangemta" (without the quotation marks)
2. Edit the following value to reflect the new location for the MTA Database Path

   HKEY_LOCAL_MACHINE \ SYSTEM \ CurrentControlSet \ Services \ MSExchangeMTA \ Parameters \ MTA Database Path
3. Explore the new path and make the new directory structure. You may omit the Mtacheck.out folder in the MTA Database Path because the MTA creates this directory automatically

   IMPORTANT: Explore the installation drive and make a backup copy of the Program Files\Exchsrvr\mtadata directory

4. Explore the original Program Files\Exchsrvr\mtadata directory
5. On the View menu, click Details, and again on the View menu, point to Arrange Icons, and then click "by Type". Select and then move all of the *.dat files to the new MTA Database path. Change the security for the directory to match that of the original directory
6. At a command-prompt, go to the Program Files\Exchsrvr\bin directory
7. In the new directory, run "mtacheck" (without the quotation marks); the task should report "Database clean, no errors detected."
8. Start the Microsoft Exchange MTA Stacks service. To do so, at the command-prompt type the following command: "net start msexchangemta" (without the quotation marks)

This procedure is covered by knowledge base article Q259896.

SMTP Transport Tuning

Mailroot directory location

When messages arrive into Exchange 2000 through the SMTP protocol, the data is written to disk in the form of an NTFS file (.EML). By default, these files are written to a directory (<drive>:\Program Files\Exchsrvr\mailroot) on the same disk partition as where the Exchange 2000 binaries are installed.

Under certain scenarios, such as configuring a bridgehead server, a positive performance impact may be noticed if the SMTP mailroot directory is located on the fastest disk partition on the computer. If it is determined that the mailroot directory is not on the most optimal disk partition, you can relocate the folder using the following steps.

1. Install the Windows 2000 Resource Kit onto your client computer (this does not need to be installed on the Exchange 2000 server)
2. Shutdown all Exchange and Internet Information Services on the Exchange 2000 server that you wish to change
3. **IMPORTANT: Explore the installation drive and make a backup copy of the Program Files\Exchsrvr\mailroot directory**
4. Move the "VSI 1" directory (and all subfolders and content) under <*drive*>:\Program Files\Exchsrvr\mailroot to the desired location. NOTE: Do not move the actual mailroot directory itself
5. Register the ADSIEDIT.DLL file using REGSVR32
6. Open a new MMC and add the ADSI Edit snap-in
7. Connect to the Configuration Container Naming Context of the Active Directory
8. Navigate to the following path:

   Configuration Container | CN=1,CN=SMTP,CN=Protocols,CN=<*server*>,CN=Servers, CN=<*admin group*>,CN=Administrative Groups,CN=<*organization*>,CN=Microsoft Exchange,CN=Services,CN=Configuration
9. Right-click on the CN=1 object and choose "Properties"
10. Change the "Select which properties to view" drop-down to "Both"
11. Adjust the paths of the following attributes (remember to click "Set" after editing):
    - msExchSmtpBadMailDirectory
    - msExchSmtpPickupDirectory
    - msExchSmtpQueueDirectory
12. Wait for Active Directory replication to replicate these changes to the rest of your forest (or at least the DC/GC servers that your Exchange 2000 computer is referencing)
13. Start the "Microsoft Exchange System Attendant" service on its own. This will copy changed paths from the Active Directory into the metabase. In less than one minute after initialisation, you should notice three 1005 events (Source: MSExchangeMU, Category: General) indicating that the paths in the metabase were updated successfully
14. Restart the Exchange 2000 server

SMTP File Handles

When the Exchange 2000 SMTP stack receives a new message, it writes the contents to a file in NTFS. Whilst the message is being processed (i.e. waiting for the next hop or delivery point) a file handle is held open with the operating system. By default, SMTP is constrained to a maximum of 1000 open file handles. This restriction is put in place to prevent out of memory problems in kernel memory and to ensure that the SMTP service shuts down in a relatively short period of time (upon shutdown, all buffers have to be flushed and file handles released).

On servers with large amounts of memory (over 1Gb) you can raise the SMTP handle threshold. Each message that is open (being processed) holds a handle and uses 5Kb of kernel memory as well as 10Kb of memory inside the INETINFO process. By raising the threshold you allow more messages to be open. This allows SMTP to process a large queue at a faster rate. However, if the total number of messages in the SMTP queues are less than 1000, this adjustment will not improve performance. Therefore only raise the value if your server is heavily loaded and you consistently see large queues.

If you increase this value then you should decrease the maximum IFS handles value to avoid running out of kernel memory in a large queue situation. When your server becomes low on kernel memory your system becomes unresponsive. To regain control of your server you will have to reboot to free up the kernel memory.

Location:	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\SMTPSVC\Queuing
Parameter:	MsgHandleThreshold (REG_DWORD)
Default setting:	Not present, but defaults to 0x3e8 (1000)
When to change:	To gain additional performance when message queues are consistently greater than 1000
Recommended setting:	Enough to accommodate the total number of messages in the queues at any one time. You should not raise the value to greater than 15000 decimal

Location:	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\SMTPSVC\Queuing
Parameter:	MsgHandleAsyncThreshold (REG_DWORD)
Default setting:	Not present, but defaults to 0x3e8 (1000)
When to change:	To gain additional performance when message queues are consistently greater than 1000
Recommended setting:	Set to the same value as “MsgHandleThreshold”

Location:	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Inetinfo\Parameters
Parameter:	FileCacheMaxHandles (REG_DWORD)
Default setting:	Not present, but defaults to 0x320 (800)
When to change:	If the “MsgHandleThreshold” registry parameter value is increased from defaults
Recommended setting:	0x258 (600)

NOTE: After adjusting the registry parameters above, you must restart the Exchange 2000 server.

SMTP Max Objects

Max Message Objects correlates to the number of messages that can be queued up at a given time by SMTP. Each mail message resident in the SMTP queue uses at least 4Kb of memory; therefore it is possible to run into a low memory situation with a very large queue. Setting Max Message Objects lower reduces the maximum number of messages that can reside in the queue, thus decreasing the maximum memory footprint for SMTP. Once this limit is reached, each SMTP connection made to the server will return with an out of memory error. For example, If this value is reduced to 10,000 Max

Message Objects, SMTP will refuse any more inbound mail once the queue reaches 10,000 messages.

Location:	HKEY_LOCAL_MACHINE\Software\Microsoft\Exchange\Mailmsg
Parameter:	MaxMessageObjects (REG_DWORD)
Default setting:	Not present but defaults to 0x186a0 (100000)
When to change:	If the Exchange 2000 server is running out of memory because the number of incoming messages is too great for the server to process

Store and ESE Tuning

Due to advances in the Store process, there is very little manual tuning that needs to be done to the Exchange store. However, additional performance can be achieved with good implementation and configuration.

Online Database Maintenance

The Store requires periodic online maintenance to be run against each database. By default, each database is set to run online maintenance between the times of 1am and 5am Online maintenance performs a variety of tasks necessary to keep the store in good health, these include, but are not limited to:

1. Checking the Active Directory to determine if there are any deleted mailboxes
2. Hard deletes any messages/mailboxes that are older than the configured retention policy
3. Online defragmentation of the data within the database

All of these operations performed by online maintenance have specific performance costs and should be understood before implementing an online maintenance strategy.

Task 1 performs an Active Directory lookup for each user in the database. The more users you have in each database, the more Active Directory (LDAP) searches will be made. These searches are used to keep the Store in sync with any Active Directory changes (specific to a check for deleted mailboxes). The performance cost of Task 1 is negligible on the Exchange server but can be significant on the Active Directory depending on the number of users, number of databases and the online maintenance times of each database. In a corporate scenario, the online maintenance occurs during the night (by default) when very few users are logged on so the load on the Active Directory servers should be very low. The extra DC load created by online maintenance should not be a problem in this scenario.

If Exchange 2000 is installed in a global data-center, serving customers from multiple time zones, then the default online maintenance time could become an issue. The effect online maintenance has on the Active Directory is proportional to the number of users in each of the server's databases. A check for a deleted mailbox is performed against each user in an database. Thus, if you have 10,000 users in a database, it will perform 10,000 LDAP searches against the Active Directory at the beginning of that databases online maintenance. If the Active Directory servers are always (around the clock) under moderate load, it would be necessary to stagger the online maintenance (set each database to start maintenance at a different time). This is especially critical if you have hundreds of thousands of users spread across dozens of servers and hundreds of databases.

Task 2 and 3 are very disk intensive and only affect the server the maintenance is being run on. During this portion of online maintenance, the server may be perceived by users as sluggish if many databases are set to perform online maintenance at the same time. Again, in corporate scenarios this would occur at night where the server can easily handle the extra load. In a global data center, it may make sense to stagger the databases (in respect to each other on a single server) to spread the disk intensive tasks over a greater amount of time.

Online backup complicates matters even further. Backing up an Exchange 2000 database will halt the maintenance of any database within that Storage Group (they will restart if the backup is finished before the maintenance interval has been passed). If you have two databases in a single Storage Group and one is running online maintenance, if a backup is started against either database, the online defragmentation on the database (which is running online maintenance) will be halted. It is critical that the backup time for any database within a Storage Group does not conflict with the maintenance times of any database within the same Storage Group. If it does, then backup will terminate the online defragmentation portion of the Store online maintenance and the database may never finish defragging.

The correct online maintenance strategy can be devised by examining the user profile (such as times of low activity), knowing how many users, databases, and servers are in the site and coordinating this information with the online backup strategy.

Here is an example of an online store maintenance schedule for a corporate Exchange 2000 mailbox server, which hosts users for a single time zone:

First Storage Group

Database One

Online maintenance runs between 9:00pm and 1:00am

Database Two

Online maintenance runs between 9:30pm and 1:30am

Database Three

Online maintenance runs between 10:00pm and 2:00am

Online Backup begins at 2:00am - backs-up all databases in First Storage Group when all databases have finished online maintenance.

Second Storage Group

Database Four

Online maintenance runs between 10:30pm and 2:30am

Database Five

Online maintenance runs between 11:00pm and 3:00am

Database Six

Online maintenance runs between 11:30pm and 3:30am

Online Backup begins at 3:30am - backs-up all databases in Second Storage Group when all databases have finished online maintenance.

This configuration will stagger the Active Directory LDAP queries generated by online maintenance which are performed in the first minutes of the procedure and ensures that online backup does not interfere with online defragmentation.

The third task of actually defragmenting the database actually consists of 18 separate sub-tasks. Once a sub-task has started it must complete. Therefore, if task 12 is still running at the end of the online maintenance window, this task will complete fully before the process exits. Therefore, online maintenance can run over the time window. Task 13 will execute during the next online maintenance window. Depending upon the run window and backup schedule, it may take a number of days before a full defragmentation completes.

Extensible Storage Engine Heaps

When Exchange 2000 is installed on servers with more than 4 processors, you may notice high virtual memory utilization by the ESE multi-heap. This can lead to performance problems especially when the server has multiple gigabytes of memory, and many databases and storage groups have been configured. It is recommended that you add the following registry parameter to all servers with more than 4 processors:

Location:	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ESE98\Global\OS\Memory
Parameter:	MPHeap parallelism (REG_SZ)
Default setting	(Doesn't exist) = Parallelism set to four times the number of processors installed
Recommended setting:	For 6 or 8 processors set the value to "0". This will hard-code the number of heaps to 11

NOTE: You must restart the Microsoft Exchange Information Store process after the above registry parameter has been adjusted.

Store Database Cache Size

Exchange 2000 ships with a hard coded maximum store database cache size. This default value is 900Mb. On servers with more than 2Gb of memory, it may be beneficial to increase the size of this cache. Due to virtual address space limitations, this value should never be set higher than 1200Mb.

NOTE: The 900Mb limit is in-place to ensure that the Store process always has ample virtual address (memory) to allocate from. Increasing this value too high can lead to system instability. See knowledge base article Q266096 for more information regarding virtual address space.

There are many factors which affect the virtual address space size in the Store.exe process. Some of these are:

- Initial allocation on start-up
- Number of Storage Groups and databases on the server
- Number of threads running
- Size of the Store Database cache

Prior to increasing the max cache size, it is recommended that the administrator use the Windows 2000 Performance Monitor to monitor the memory of the server under normal load. Monitor the following performance monitor value:

Performance Object: Process

Counter: Virtual Bytes

Instance: STORE

This will give you an accurate value for the virtual address space the Store has allocated. On a server with the /3GB setting in the boot.ini, this should be below 2.8Gb. On a server without the /3GB setting in the boot.ini, this should be below 1.8Gb (it is recommended that servers with 1Gb or more memory have the /3GB switch added to boot.ini). If you see values that are higher than this for either configuration, do not increase the size of your max cache size. If you see values that are lower than this for either configuration, then you can safely increase the size of your database max cache size. I.e. If you have a /3GB configured server and performance monitor shows the virtual bytes count at 2.5Gb under heavy load, then you know you are safe to increase your max cache size by 300Mb, or 1200Mb total.

Steps to modify msExchESEParamCacheSizeMax:

This procedure relies on the use of the ADSI Edit tool. This is included with the Windows 2000 Resource Kit. Once you have installed the Resource Kit, register the tool using the following command:

```
REGSVR32 ADSIEDIT.DLL
```

To use the ADSI Edit tool, launch a new MMC process (from Start | Run) and add the “ADSI Edit” snap-in to the console. In the main console window, right-click on the ADSI Edit node and choose “Connect to”. When the Connection dialog box appears, change the Naming Context to “Configuration Container”. For more information on ADSI Edit refer to the Windows 2000 Resource Kit documentation.

1. Using ADSI Edit, connect to the Configuration Container Naming Context of your Active Directory
2. Navigate to the following path: Configuration Container | CN=Information Store,CN=<*server*>,CN=Servers,CN=<*Admin Group*>,CN=Administrative Groups,CN=<*org*>,CN=Microsoft Exchange,CN=Services,CN=Configuration

3. Right-click on the Information Store object and select 'Properties'
4. Change the "Select which properties to view:" drop-down to "Both"
5. Select the "msExchESEParamCacheSizeMax" attribute and adjust the value. Although no value will be present, the default will be 230400 (which is 900Mb). The recommended maximum for this value is 307200 (which is 1200Mb). **NOTE:** Be careful when setting this value, as it is very easy to make a mistake and set the "msExchESEParamCacheSize***Min***" attribute instead.
6. Remember to click "Set" after changing the edit field for the attribute
7. Close down the ADSI Edit tool by closing the MMC console application
8. Wait for Active Directory replication to replicate this new value throughout the forest
9. Restart the Information Store service on the Exchange 2000 server

Log Buffers

ESE uses a set of log buffers to hold information in memory before writing to the transaction logs. For back-end servers, the default value is too low. This can cause excessive disk I/Os to the transaction log drive. A significant performance improvement will be seen if the server is under load or if users are sending large messages. The default value is 84 – this should be increased to 9000 on all servers.

The process for setting Log Buffers is very similar to increasing the Store Database Cache Size (as detailed above). You will need to use ADSI Edit to navigate to the Storage Group object and then find the "msExchESEParamLogBuffers" attribute. The value is an integer and it should be set manually to 9000.

Active Directory Connector Tuning

The ADC requires very little tuning during normal operation. There are only really two scenarios where you might want to consider manually tuning the ADC process.

Sleep time – Once the ADC has fully replicated Exchange and Active Directory data, it simply performs replication on the changes made to those directories, and in the majority of circumstances, those deltas will be small. During a connection agreement's activation time, the ADC is permitted to work continuously for 5 minutes. After that, the ADC will "sleep" for 5 minutes to allow other applications, including Active Directory's own replication mechanism processing time on the DC/GC server. However, if a connection agreement is executing for the first time, or if a large number of changes are made to one of the directorys, you may want to permit the ADC to perform the replication without sleeping (and thus speed up the replication cycle). You can configure the maximum time that the ADC is permitted to work without sleeping and the maximum time the ADC should sleep. You must understand that making these changes will effect all of the connection agreements running on that ADC server, and that you may adversely affect other Active Directory applications.

Location:	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\MSADC\Parameters
Parameter:	Max Continuous Sync (secs) (REG_DWORD)
Default setting:	Not present , but will default to 0x12c (300)
When to change:	If you wish the ADC to continue processing even after replicating solidly for 5 minutes
Recommended setting:	Set to no more than 20 minutes (1200 seconds). Setting this value too high can have an adverse effect on the Active Directory and other applications

Location:	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\MSADC\Parameters
Parameter:	Sync Sleep Delay (secs) (REG_DWORD)
Default setting:	Not present , but will default to 0x12c (300)
When to change:	If you wish to adjust the default sleep time of the ADC
Recommended setting:	Set to a minimum of one minute (60 seconds). Wherever possible, leave this setting at default. Be careful when raising this value. For example, if you set this to 3600 seconds (1 hour) and set the Connection Agreement replication schedule for individual 15 minute segments, the ADC may never replicate. It is recommended that you only change the sleep delay when setting to replication schedule to 'Always'

More information on this subject can be found in knowledge base article Q253825

Block searching – By default, the ADC requests changes from the Exchange and Active Directory in blocks of 10,000. If there are over 10,000 objects to be replicated, the ADC will request the first 10,000 entries, process them, and ask for the next 10,000. When the ADC is communicating with directory servers over an error-prone network it may be useful to reduce the block size. The reason for this is that if the ADC receives a partial block, and the connection to the directory service is lost, the complete block will need to be re-replicated when the connection is re-established.

Location:	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\MSADC\Parameters
Parameter:	Export Block Size (REG_DWORD)
Default setting:	Not present , but will default to 0x2710 (10000)
When to change:	If you need the ADC to commit before 10,000 new or changed objects have been received. You would generally lower this value in difficult WAN situations
Recommended setting:	Set as appropriate. Be careful not to set the value too low or you will receive poor performance from the ADC

More information on this subject can be found in knowledge base articles Q253665 and Q253840.

Active Directory Integration Tuning

Increasing the maximum active LDAP queries on Active Directory servers

When there are numerous Exchange 2000 servers in a Windows 2000 site, a very large LDAP load could be put on the Active Directory servers. An Active Directory serve, by default, is configured to support a maximum of 20 active LDAP queries. If this limit is reached, the Active Directory will return the error 'LDAP_ADMIN_LIMIT_EXCEEDED' and will refuse processing further LDAP queries. 20 is generally sufficient for most Active Directory servers, but it is necessary to increase this value when you are running Exchange 2000 on 6 or 8 processor servers, or if the above error message is logged.

The maximum LDAP queries can be configured through the "MaxActiveQueries" attribute. This can be adjusted using the NTDSUTIL.EXE tool. Increasing this setting will use a little more memory in the LSASS.EXE process on the Active Directory server, so do not increase this value any higher than is necessary. The steps for changing this value are as follows:

1. Start NTDSUTIL
2. Type LDAP POLICIES
3. Type CONNECTIONS
4. Type CONNECT TO SERVER <dc/gc name>
5. Type Q
6. Type SHOW VALUES
7. Type SET MAXACTIVEQUERIES TO 40
8. Type COMMIT CHANGES
9. Type SHOW VALUES
10. Verify that the new setting is shown
11. Type Q
12. Type Q

NOTE: This setting will be replicated to all Active Directory servers within the **forest**. You do not have to reboot DC/GC servers for this to take effect.

Outlook Web Access Tuning

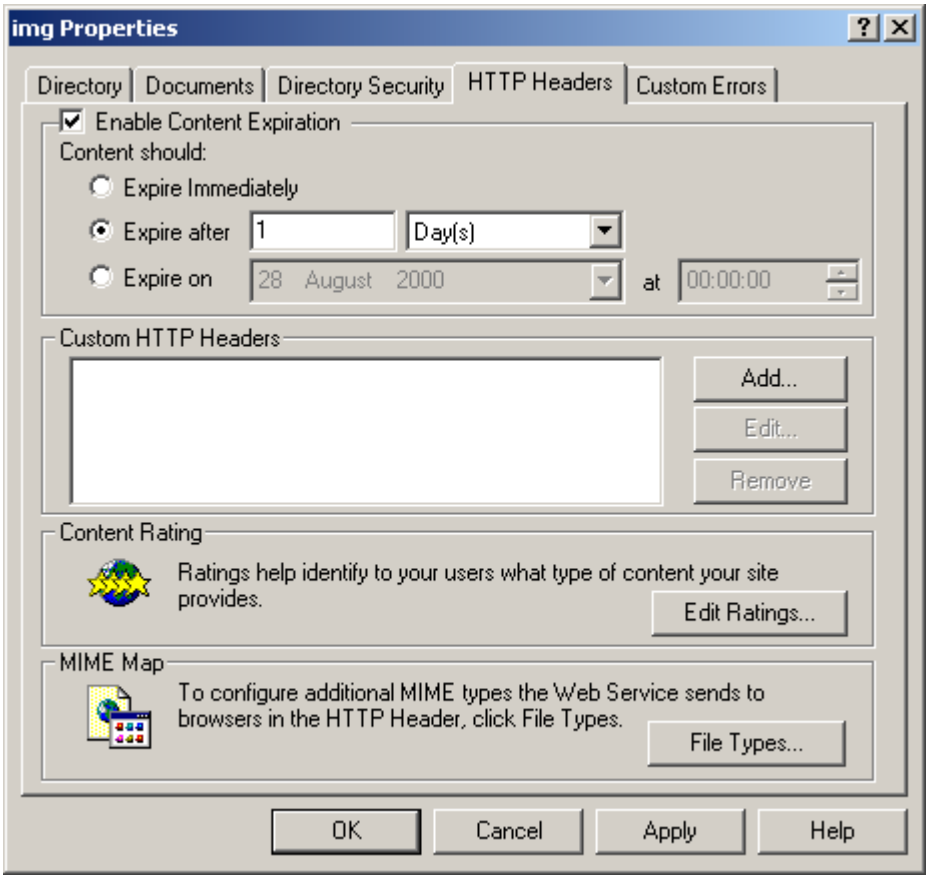
Content Expiration

Outlook Web Access (OWA) permits users to access their mailboxes over the Internet or an intranet using a regular browser such as Internet Explorer 5.x. For Exchange 2000 servers on which many or most of the users will have OWA as their client, traffic and resources on the server – as well as latencies apparent to the user – can all be reduced by increasing the time until expiration of static files from the *\Program Files\Exchsrvr\exchweb\controls* and *\img* directories. These static files include all the *.gif* icons, navigation or tool bars, and windows for display of messages, calendar items, etc.

By default, these files are marked as expiring in one day when they are received from the server by the browser; thus, once each day, each user will request and receive these files from the server. Except in cases where "static" content is modified to include system messages or advertisements, these files will not change except possibly in the event of a future Exchange Service Pack installation. Increasing the time before expiration to a month or a year (or setting the content to not expire) may make more sense. In such a case, these objects might only be expired out of the client's browser cache if the cache became full and older items were deleted. Additionally, each time the user accesses Outlook Web Access, the browser will send an "is modified" request for each static file cached, therefore, even if a future Service Pack does modify *\exchweb* content, the browser will immediately pick up the changes.

If you ever need to force the browser to refresh the entire content on a page, thus ignoring data in the cache, open the URL, hold down CTRL + SHIFT and then click the Refresh button.

Changing this setting does not involve the Registry; instead, launch the **Internet Information Services Manager**, double-click to expand the server icon, expand the **Default Web Site**, expand **Exchweb**, and view Properties on the **\controls** and **\img** directories. Under the **HTTP Headers** tab, you can either unselect the Enable Content Expiration checkbox, or set the Expire after property to a value greater than the one day default.



Cluster Performance Tuning

There are five work queues in SMTP that use a pool of threads known as the ATQ threads. By design, two of these queues can, by default, use up to 90% of the available threads. This thread pool is shared by the process accepting POP and IMAP requests. Therefore, in a moderate to high load scenario, a situation can arise where SMTP can starve resources from POP and IMAP.

We can reserve adequate threads for POP and IMAP by limiting the percentage of threads SMTP can use. To accomplish this we have to increase the overall number of threads available to IIS. The only tradeoff here is increased memory usage for the additional threads.

There are two registry values that allow you to control the "SMTP % of threads" and "Additional threads per processor".

Location:	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\SMTPSVC\Queuing
Parameter:	MaxPercentPoolThreads (REG_DWORD)
Description:	Maximum Percentage of ATQ Threads that each Queue will request
Default setting:	Not present , but will default to 0x5A (90)
When to change:	High SMTP activity causing the POP and IMAP instances to fail in the Cluster Administrator MMC
Recommended setting:	Use the following formula to determine the settings: MaxPercentPoolThreads = 90/(2 * Number Of Virtual Server Instances). Example: Assuming a cluster with 2 VSIs = 90/(2*2) = 22. If you get a decimal point in the calculation, please round down to the next integer

Location:	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\SMTPSVC\Queuing
Parameter:	AdditionalPoolThreadsPerProc (REG_DWORD)
Description:	Additional pool per processor threads when SMTP is started
Default setting:	Not present , but will default to 0x6 (6)
When to change:	High SMTP activity causing the POP and IMAP instances to fail in the Cluster Administrator MMC. Change in conjunction with MaxPercentPoolThreads
Recommended setting:	Use the following formula to determine the settings: AdditionalPoolThreadsPerProc = ((9/ (MaxPercentPoolThreads/100)) – 4) / 2. Example: Assuming a cluster with 2 VSIs = ((9/(0.22))-4)/2 = (41-4)/2 = 18. If you get a number greater than 200, you need to assign that number to the following registry value: HKLM\ System\CurrentControlSet\Services\InetInfo\Parameters\PoolThreadLimit

Acknowledgements

This document could not have been produced without the help and expertise from the following people:

Milan Shah – Exchange 2000 Transport Developer

Ted Dame – Exchange 2000 Transport Test Team

Matt Gossage – Exchange Performance and Scalability Team

Sean O'Brien – Exchange Performance Modeling Team

Kevin Bonilla – Exchange Performance Modeling Team

Curtis Manley – Exchange Performance Modeling Team

Jaime Ondrusek – Exchange Performance Modeling Team

Paul Appleby – European Regional Support Center (Exchange)

Scott Oseychik – Exchange 2000 Beta Support Team

Channing Heffney – Exchange 2000 Beta Support Team

Bryan Hunt – Exchange 2000 Beta Support Team

Kent Tilger – Exchange 2000 Beta Support Team

Eddie Fong – Exchange 2000 Beta Support Team

John Kenerson – Exchange 2000 JDP Management Team

Finbar Ryan – European Regional Support Center (IIS)